# Scheduling Optimization on the Simbus Backplane

Dale Martin, Philip Wilsey - Clifton Labs, Inc.

Robert J. Hoekstra, Eric R. Keiter,

Scott A. Hutchinson, Thomas V. Russo, and
Lon J. Waters - Sandia National Labs

Clifton Labs, Inc.

# Presentation Outline

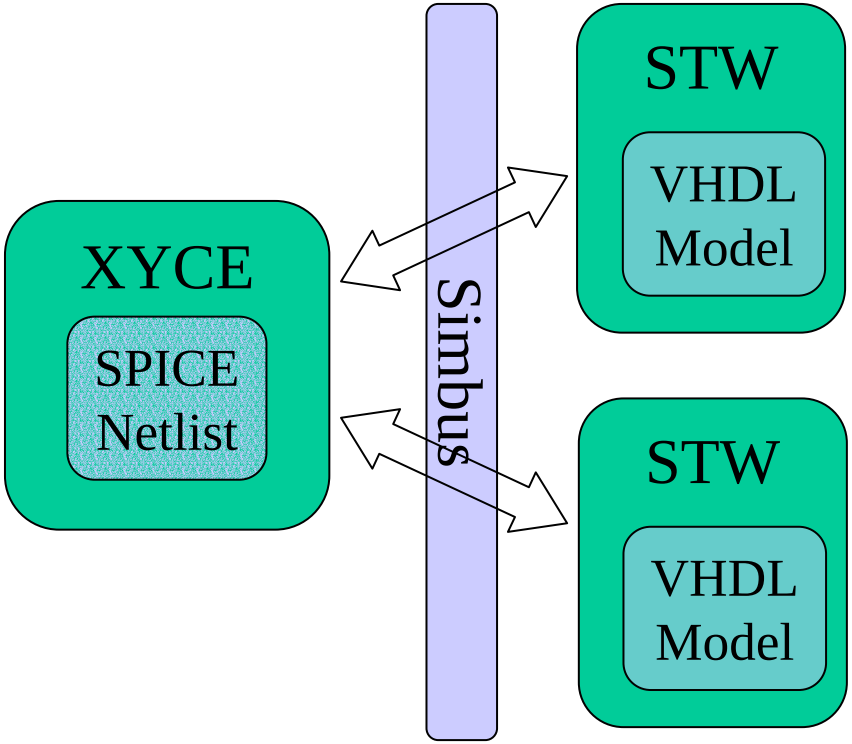
- Quick overview of the Simbus project
- Fixed time-step algorithm and it’s problems
- Variable time-step scheduling algorithm
- Results
- Conclusion

# Simbus Project Goals

- Enable parallel mixed-signal simulation
  - Leverage XYCE and Savant/Tyvis/Warped (STW)
  - Leverage existing models
    - Rules out VHDL-AMS
  - Leverage parallel computing infrastructure

# $0.25 Tour of Simbus

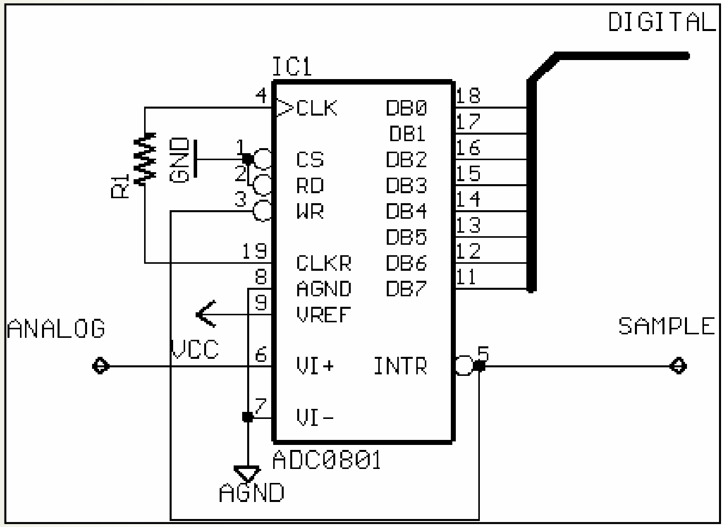
- Backplane Solution
  - XYCE executes SPICE models
  - STW executes VHDL models
  - Simbus is the glue
    - Does the scheduling
    - Delivers events



Clifton Labs, Inc.

# What are the domain boundaries?

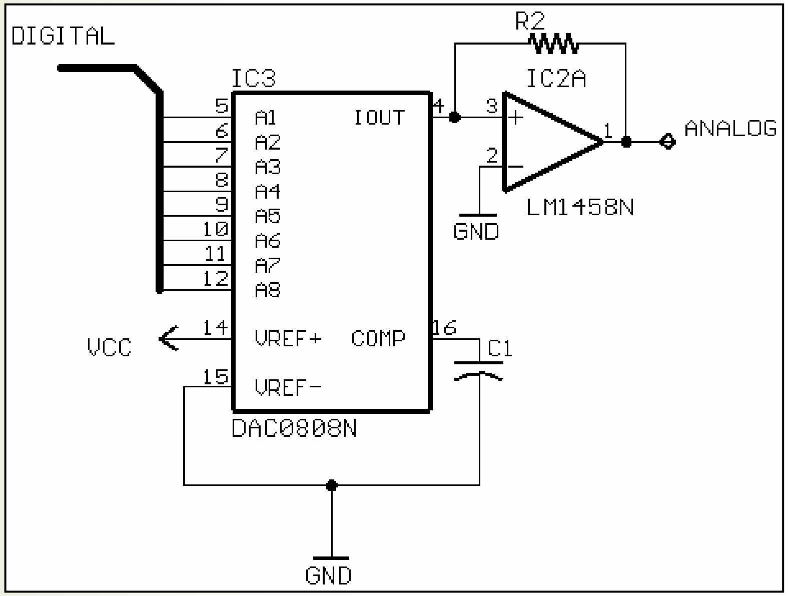
- What are they in “real life”?
  - A/D converters



Clifton Labs, Inc.

# What are the domain boundaries? (2)

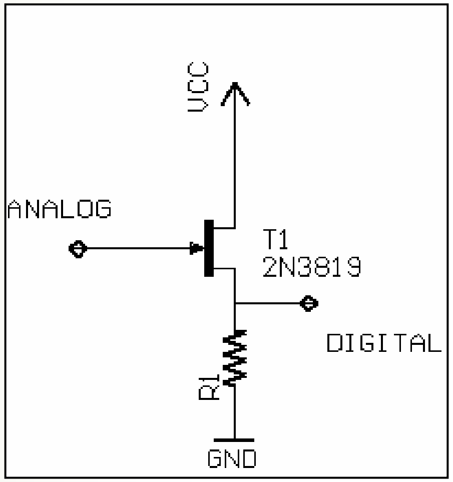
- What are they in “real life”?
  - D/A converters



Clifton Labs, Inc.

# What are the domain boundaries? (3)

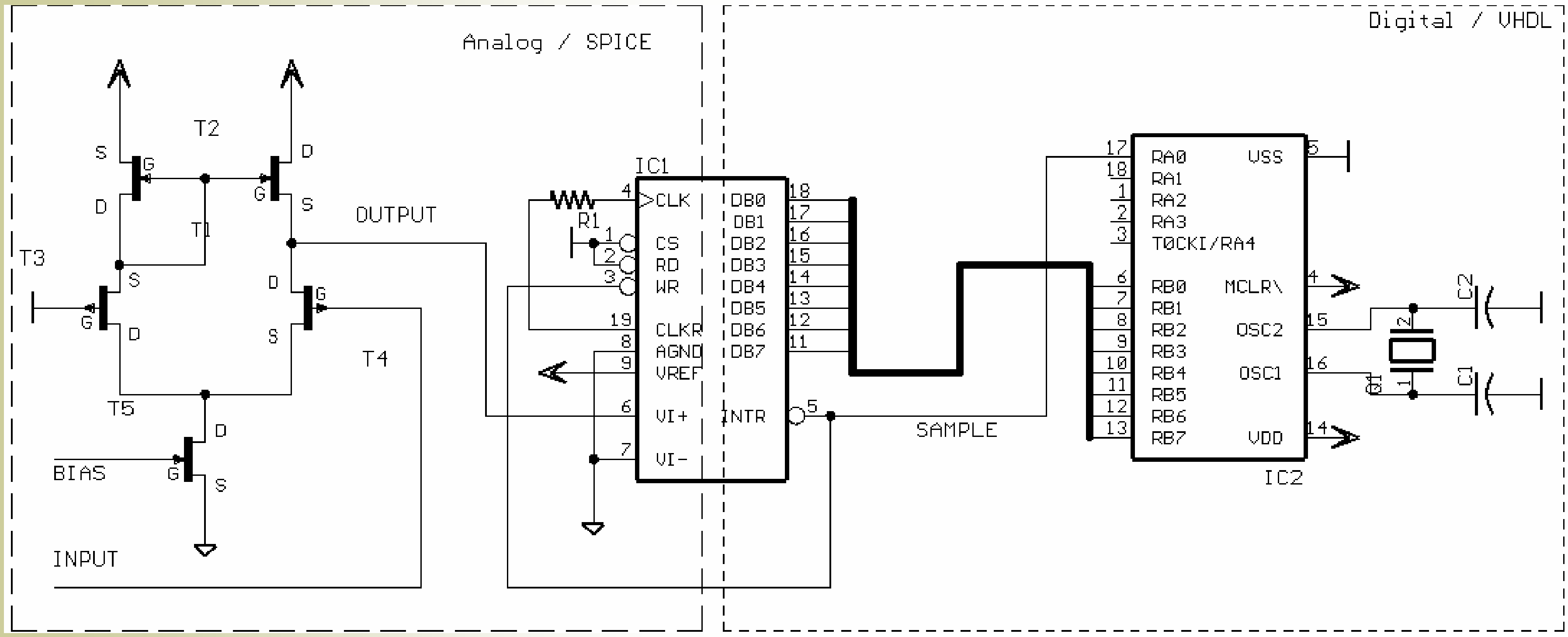
- What are they in “real life”?
  - Other circuits acting as one or the other



# Modeled domain boundaries

- Explicit models for domain crossing devices.
  - – Currently A/D and D/A devices.
  - – More models could be produced for domains in the “other” category.
- Model instances are required in both SPICE and in VHDL.

# Example


Analog / SPICE
Digital / VHDL
T1
T2
T3
T4
T5
S
G
D
OUTPUT
BIAS
INPUT
R1
IC1
>CLK
CS
RD
WR
CLKR
AGND
VREF
VI+
VI-
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
INTR
SAMPLE
IC2
RA0
RA1
RA2
RA3
T0CKI/RA4
RB0
RB1
RB2
RB3
RB4
RB5
RB6
RB7
USS
MCLR\
OSC2
OSC1
VDD
C1
C2
Q1
4
1
2
3
19
8
9
6
7
18
17
16
15
14
13
12
11
5
10

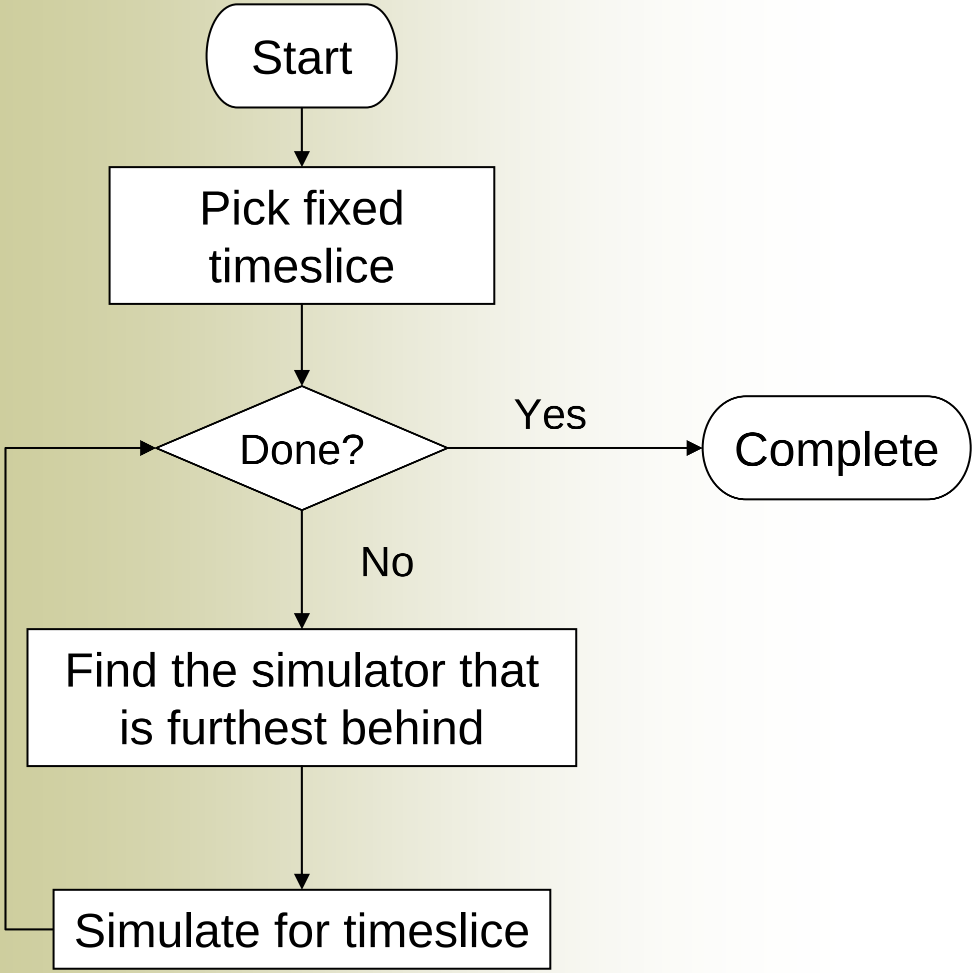
# Backplane “Glue”

```
PluginDir: "/opt/simulation/simbus/plugins"

Simulators {
  TyvisSimulation {
    SimulationPlugin : “a2d-master.la"
  }

  XyceSimulation {
    NetList : “diff-amp-test.ckt”
  }
}
```

# Fixed Time-Step (FTS) Scheduling

- Requirements
  - Non-zero latencies not allowed across backplane
  - Fixed time-step
    - Bounded by minimum conversion time
    - => Fixed minimum conversion time for all domain boundaries

# FTS Scheduling (2)



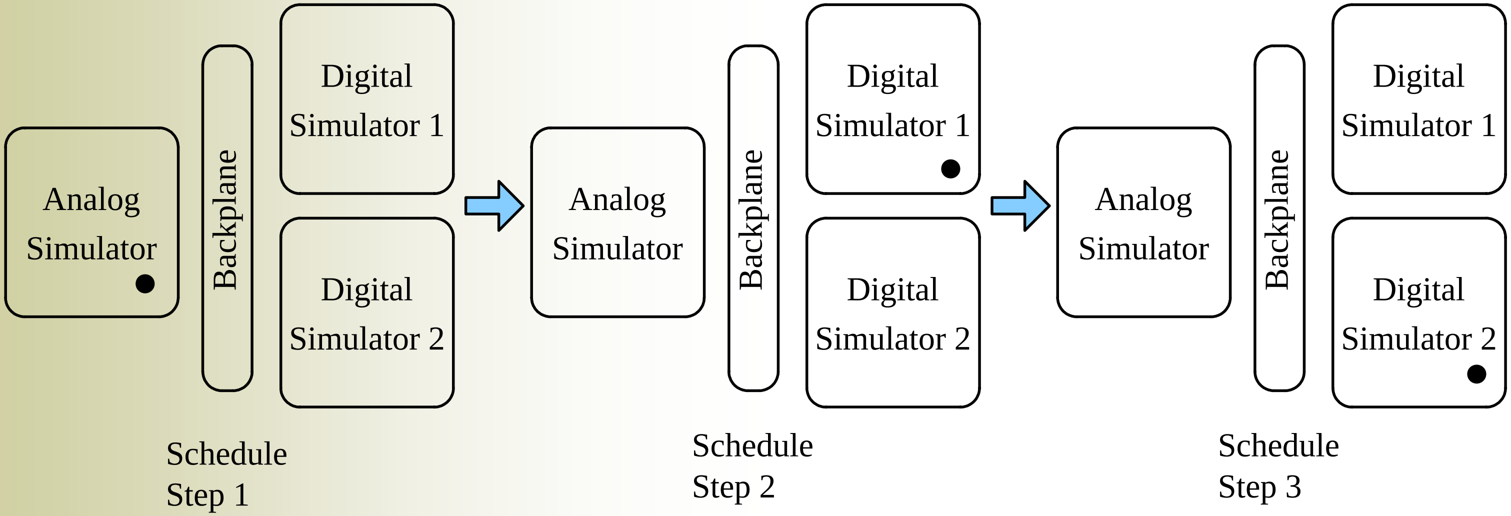
Analog Simulator
Committed Time: 0ns

Digital Simulator 1
Next event: 250 ns
Committed Time: 0 ns

Digital Simulator 2
Next event: 350 ns
Committed Time: 0 ns

CLIFTON LABS, INC.

# FTS Scheduling (3)



Analog Simulator ●

Backplane

Digital Simulator 1

Digital Simulator 2

Schedule Step 1

Analog Simulator

Backplane

Digital Simulator 1 ●

Digital Simulator 2

Schedule Step 2

Analog Simulator

Backplane

Digital Simulator 1

Digital Simulator 2 ●

Schedule Step 3

Clifton Labs, Inc.

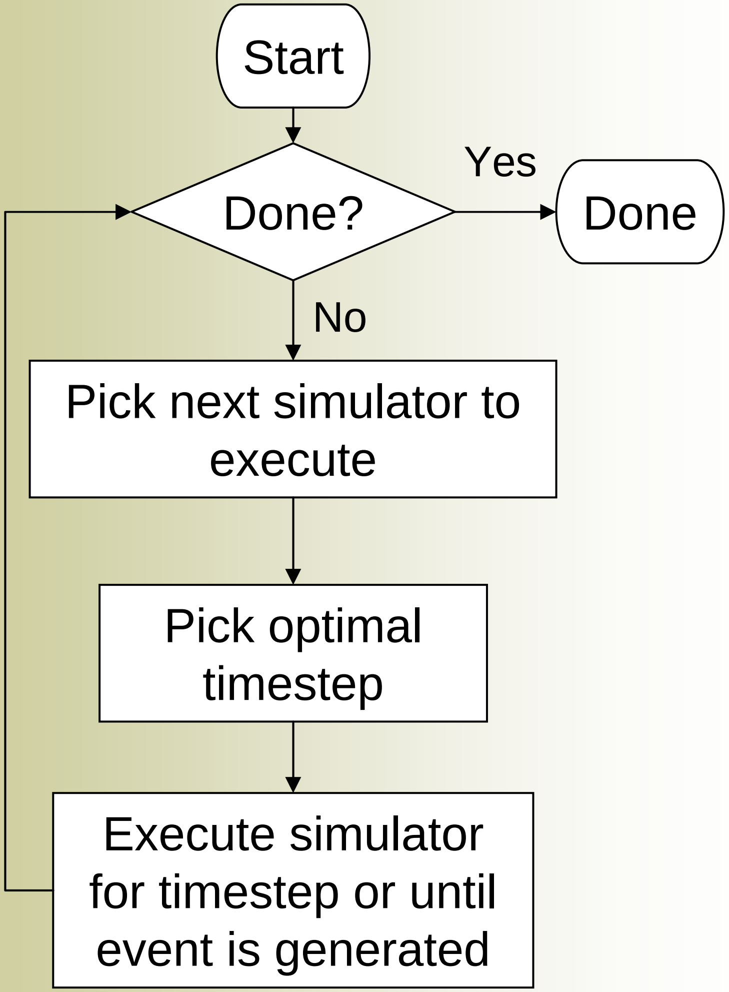
# FTS Scheduling (4)

- The problems with FTS scheduling:
  - Inefficient, especially if we have low latency domain converters (e.g. FETs/transistors)
  - E.g. fixed step could be on the order of nanoseconds, where our system clock is at tens or hundreds of nanoseconds

# Variable Time-Step (VTS) Scheduling

- How can we pick a bigger time-step?
  - Maximum time-step == time until next backplane crossing event
  - We can get this for the event driven simulators
  - We cannot know this from the continuous simulator

# VTS Scheduling (2)



Start

Done?

Yes

Done

No

Pick next simulator to execute

Pick optimal timestep

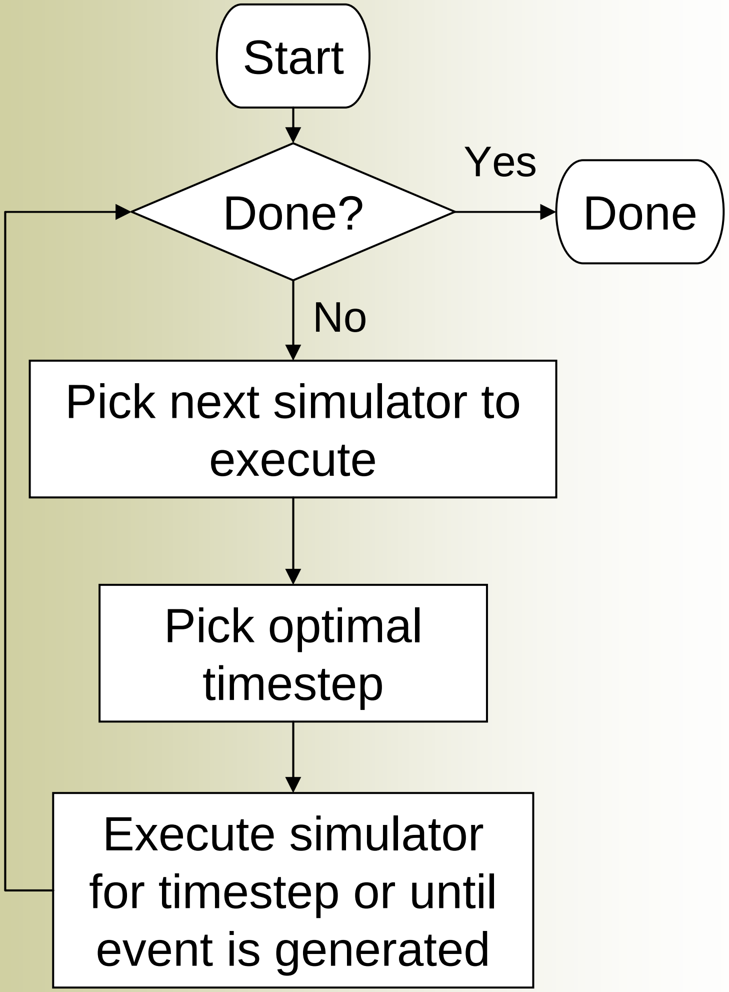
Execute simulator for timestep or until event is generated

Analog Simulator
Next Event: ??
Committed Time: 0ns

Digital Simulator 1
Next event: 250 ns
Committed Time: 0 ns

Digital Simulator 2
Next event: 350 ns
Committed Time: 0 ns

# Execution, No Event Generated

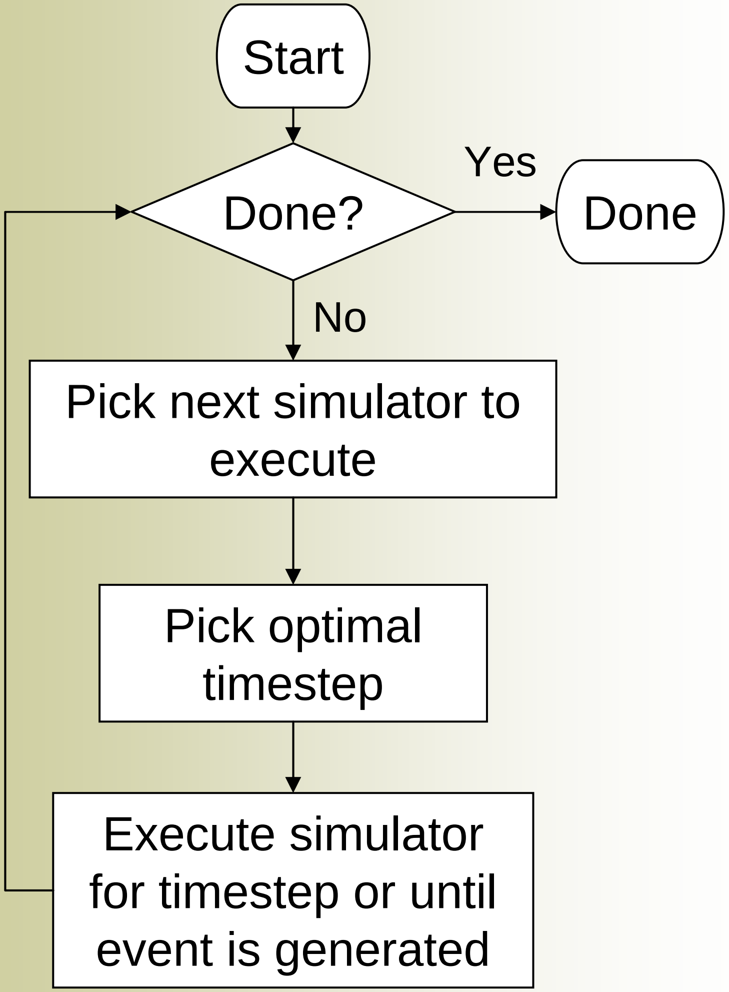

Start
Done?
Yes
Done
No
Pick next simulator to execute
Pick optimal timestep
Execute simulator for timestep or until event is generated

Analog Simulator
Next Event: ??
Committed Time: 250 ns

Digital Simulator 1
Next event: 250 ns
Committed Time: 0 ns

Digital Simulator 2
Next event: 350 ns
Committed Time: 0 ns

# Execution, Event Generated


Start
Done?
Yes
Done
No
Pick next simulator to execute
Pick optimal timestep
Execute simulator for timestep or until event is generated

Analog Simulator
Next Event: ??
Committed Time: 150 ns

Digital Simulator 1
Next event: 250 ns
Committed Time: 0ns

Digital Simulator 2
Next event: 170 ns
Committed Time: 0 ns

Clifton Labs, Inc.

# Results

| | FTS Scheduler | | | VTS Scheduler | | |
|---|---|---|---|---|---|---|
| | ADC Example | DAC Example | U-Shaped Example | ADC Example | DAC Example | U-Shaped Example |
| Mean Execution Time | 0.16s | 7.52s | 5.9 | 0.12s | 9.03 | 4.03 |
| Speedup | | | | 24.05% | -20.05% | 32.30% |
| Scheduling Overhead | 0.33% | 16.22% | 13% | 1.5% | 0.38% | 0.92% |
| Digital Contribution | 73.4% | 77.97% | 71.77% | 70.92% | 98.36% | 87.26% |
| Analog Contribution | 8% | 0.30% | 3.0% | 7.29% | 0.53% | 4.58% |
| Total Backplane Overhead | 18.56% | 21.73% | 25.23% | 21.79% | 1.11% | 8.16% |

# Conclusions

- Efficient scheduling is important in mixed-signal systems
- Fixed time steps will not work well in general
- Variable time-step scheduling can improve performance